SR56x0 Product Errata

Silicon Errata for SR5690, SR5670 and SR5650

Publication #	**46303** Revision: **3.00**
Issue Date:	**July 2011**

Trademarks

Revision History

Date	Revision	Description
July 2011	3.00	• Initial public release

Product Errata Summary

Except where otherwise noted, this product errata is applicable to SR5690, SR5670 and SR5650.

A unique errata reference number (ERN) has been assigned to each erratum within this document for user convenience in tracking the errata within specific revision levels. Table 1 cross-references the revisions of the part to each erratum. An "X" indicates that the erratum applies to the revision. The absence of an "X" indicates that the erratum does not apply to the revision. An "*" indicates advance information that the erratum has been fixed but not yet verified. "No fix planned" indicates that no fix is planned for current or future revisions of the ASIC.

Note: *There may be missing errata numbers. Errata that have been resolved from early revisions of the ASIC have been deleted, and errata that have been reconsidered may have been deleted or renumbered.*

Table 1: Cross-Reference of Product Revision to Errata

#	Errata Description	ASIC Revision
		A21
1	IOMMU L1 Interrupt Identification	No fix planned
2	Non-optimal HTIU Splitting of Host Requests	No fix planned
4	HT Compat Bit Set in Upstream Isoc DMA	No fix planned
6	Downstream Stomped HyperTransport™ Responses	No fix planned
9	Glitch on the HyperTransport Interface When Using LS1 Combined with the PHY_OFF Inactive Lane State when Increasing the HyperTransport Link Width	No fix planned
10	IOMMU Event Log Consistency	No fix planned
12	Attributes for "Special" Upstream TgtDone Responses may be Incorrect	No fix planned
30	Certain PCIe® Msg Packets are not Logged as Unsupported Request (UR) Errors	No fix planned
49	LDTSTOP Assertion During HyperTransport Link Training May Lead to an Incorrect Link State Transition	No fix planned
56	HyperTransport Clock Data Recovery (CDR) Disabled Too Late When Changing Link Width Using LDTSTOP	No fix planned
58	IOMMU May Pass Reads in the HyperTransport System Management Address Range	No fix planned
60	Zero-Byte DMA Reads are Sent to Physical Address 0x0	No fix planned
61	IOMMU Aborts ATS Address Translation Requests to the Exclusion Range when DTE.TV=0	No fix planned
62	HyperTransport BIST ErrLnNum Priority is Incorrect	No fix planned
63	IOMMU Blocks Writes in the HyperTransport System Management Address Range when SysMgt=0x1	No fix planned
64	Certain IOMMU Registers Not Initialized/Cleared Properly	No fix planned
66	ECRC Check Enable and ECRC Generation Enable Registers Clear on Warm Reset.	No fix planned
67	Replay Timer Timeout Status Set Incorrectly	No fix planned
68	Error Containment – HyperTransport Interface	No fix planned
71	Loss of Flow Control During LDTSTOP# When Using Hypertransport Gen1 Mode	No fix planned
72	DMA Request May Be Incorrectly Logged As Unsupported Request In PCIe Error Reporting Registers	No fix planned
73	PCIe Advanced Error Reporting (AER) Header Log May Record Incorrect Information	No fix planned
75	Incorrect Check of PCIe ECRC on Specific Device Configurations of x8 and x16 Links	No fix planned
76	Read Completion Timeout with Relaxed Ordering	No fix planned
77	Read Completion Timeout	No fix planned

Table 1 (continued): Cross-Reference of Product Revision to Errata

#	Errata Description	ASIC Revision
		A21
79	IOMMU Event Log Ordering Violation	No fix planned
80	HTIU False Parity Errors	No fix planned

Product Errata

1 IOMMU L1 Interrupt Identification

Description

IOMMU L1 identifies all requests <= 1DW in size in the interrupt ranges (HyperTransport™ or MSI) as interrupts when real interrupts should only be 1DW in size.

Potential Effect on System

Malformed interrupt requests (not all byte enables set) may be treated by IOMMU as normal interrupts.

Suggested Workaround

None, as these types of requests are not expected during normal operation.

Fix Planned

No

2 Non-optimal HTIU Splitting of Host Requests

Description

HTIU will split host requests that are not naturally aligned but may do so in a non-optimal way.

Potential Effect on System

Lower performance for requests that are not naturally aligned.

Suggested Workaround

None

Fix Planned

No

4 HT Compat Bit Set in Upstream Isoc DMA

Description

The chipset sets the compat bit whenever it sets the Isoc bit for upstream DMA.

Potential Effect on System

None. The processor ignores the compat attribute in upstream DMA requests.

Suggested Workaround

None

Fix Planned

No

6 Downstream Stomped HyperTransport Responses

Description

Downstream stomped read responses on the HyperTransport interface may result in data corruption.

Potential Effect on System

None, as it has been confirmed that the processor will not send this type of request.

Suggested Workaround

None

Fix Planned

No

9 Glitch on the HyperTranport Interface When Using LS1 Combined with the PHY_OFF Inactive Lane State when Increasing the HyperTransport Link Width

Description

Under this condition (PHY_OFF inactive lane state and LS1), the HyperTransport transmitter is reset to realign the active and inactive lanes when doing an HyperTransport link width increase. However, given that the clock lanes are also reset, a glitch is observed on the clock.

Potential Effect on System

Cannot support this particular configuration (PHY_OFF inactive lane state and LS1).

Suggested Workaround

Either set the inactive lane state to “operational” while continuing to support LS1 or configure HyperTransport to use LS2 while continuing to support the PHY_OFF inactive lane state.

Fix Planned

No

10 IOMMU Event Log Consistency

Description

When the IOMMU detects specific combinations of errors within the same transaction, the resulting IOMMU event log entry may contain event information in reserved bits that should have only been set for another event log type. These reserved bits should be ignored by software. The Device ID, Address fields, and defined status bits are correct.

Potential Effect on System

IOMMU event logs may be generated that do not strictly correspond to any of the formats defined in the IOMMU specification.

Suggested Workaround

None

Fix Planned

No

12 Attributes for “Special” Upstream TgtDone Responses May Be Incorrect

Description

When HTIU converts a non-posted host write to a posted one, it internally creates a TgtDone response. This response may have PassPW attributes set incorrectly.

Potential Effect on System

None, as TgtDone responses that may have the incorrectly set PassPW attributes have no ordering requirements relative to other packets.

Suggested Workaround

None

Fix Planned

No

30 Certain PCIe® Msg Packets are not Logged as Unsupported Request (UR) Errors

Description

UR is not reported for unsupported upstream type 0 Vendor Defined Messages and for messages with invalid message codes.

Potential Effect on System

Error status may not be logged if the specified message types (i.e., type 0 Vendor Defined Messages) are received. Such messages should not be received during normal operation.

Suggested Workaround

None

Fix Planned

No

49 LDTSTOP Assertion During HyperTransport Link Training May Lead to an Incorrect Link State Transition

Description

If LDTSTOP is asserted while the north bridge HyperTransport receiver is in training 2 state and the HyperTransport transmitter is in training 3 state, the transmitter does not proceed to operational or send 200ns of disconnect NOPs before shutting off the link. Instead, the transmitter sends training patterns for 200ns before shutting off the link.

Potential Effect on System

None expected.

Suggested Workaround

None required.

Fix Planned

No

56 HyperTransport Clock Data Recovery (CDR) Disabled Too Late When Changing Link Width Using LDTSTOP

Description

When decreasing HyperTransport link width using LDTSTOP, the PHY receiver CDR is disabled too late. The CDR may pick up garbage samples and drift slightly off the nominal position.

Potential Effect on System

HyperTransport training may take slightly longer when doing a width increase using LDTSTOP.

Suggested Workaround

None, as HyperTransport width changes are not made during normal system operation.

Fix Planned

No

58 IOMMU May Pass Reads in the HyperTransport System Management Address Range

Description

Under a highly specific sequence of requests and internal timing conditions, the IOMMU may pass through read requests in the HyperTransport system management address range when the associated device table entry has the SysMgt field set to 01b (pass through writes). These are erroneous requests since devices should not be allowed to issue reads to this address range when SysMgt=01b.

Potential Effect on System

Unpredictable system behaviour. This issue has only been observed in the simulation environment.

Suggested Workaround

Only set DTE.SysMgt=01b for the south bridge ACPI controller. The south bridge is the only device that should be generating requests in the HyperTransport interrupt address range and it is not expected to propagate read requests in this range.

Fix Planned

No

60 Zero-Byte DMA Reads are Sent to Physical Address 0x0

Description

All zero-byte DMA reads are sent to physical address 0x0.

Potential Effect on System

None

Suggested Workaround

None

Fix Planned

No

61 IOMMU Aborts ATS Address Translation Requests to the Exclusion Range when DTE.TV=0

Description

IOMMU master aborts ATS address translation requests to the exclusion range when the associated device table entry has DTE.TV=0 and either DTE.EX=1 or the EX_ALLOW register is set to 1. Having DTE.TV=0 implies that the device does not have any page tables and is only issuing DMA requests into the exclusion range.

Potential Effect on System

Endpoint devices are prevented from issuing translated requests to the exclusion range under the described device table entry configuration.

Suggested Workaround

Set DTE.TV=1, DTE.IR=0 and DTE.IW=0.

Fix Planned

No

62 HyperTransport BIST ErrLnNum Priority is Incorrect

Description

The HyperTransport specification defines the BIST ErrLnNum register to record the highest numbered lane with an error but the device reports the lowest numbered lane with an error.

Potential Effect on System

None, however, this is an HyperTransport specification violation.

Suggested Workaround

None

Fix Planned

No

63 IOMMU Blocks Writes in the HyperTransport System Management Address Range when SysMgt=0x1

Description

Write requests to the HyperTransport system management address range may be blocked by the IOMMU when the associated device table entry has the SysMgt field set to 01b (pass through writes) with V=1, TV=1 and IW=0.

Potential Effect on System

Unpredictable system behaviour. This issue has only been observed in the simulation environment.

Suggested Workaround

Only set DTE.IW=1 for the south bridge ACPI controller as this is the only device in the system that should issue system management requests.

Fix Planned

No

64 Certain IOMMU Registers Not Initialized or Cleared Properly

Description

The IOMMU command buffer and event log tail and head pointers are only reset on a cold boot rather than warm boot and are not reset when the command buffer base register or event log base register are written as per the IOMMU specification.

Potential Effect on System

The IOMMU command buffer tail pointer and event log head pointer are not set to zero when exiting a warm reset or after the associated base address registers are written. This may lead to commands being fetched or events being indicated once either of these features are enabled. Additionally, reading the IOMMU command buffer head pointer or event log tail pointer registers after a reset but before the pointers are explicitly written and before the associated base address registers are written may return an unpredictable value.

Suggested Workaround

The System BIOS should explicitly clear the IOMMU pointer registers on every reset. IOMMU software should explicitly clear the IOMMU pointer registers after writing the command buffer and event log base address registers before enabling the associated feature.

Fix Planned

No

66 ECRC Check Enable and ECRC Generation Enable Registers Clear on Warm Reset

Description

The ECRC Check Enable and ECRC Generation Enable registers in the PCIe Advanced Error Capabilities and Control register are defined as sticky in the PCIe specification. PCIe sticky register states are intended to be preserved across a warm reset and cleared on cold reset. The affected ECRC Enable registers clear on warm reset instead of cold reset.

Potential Effect on System

ECRC features are disabled after every warm reset. There is no impact to link traffic since PCIe links are disabled and need to be re-initialized.

Suggested Workaround

System BIOS should reprogram the affected registers after every warm reset.

Fix Planned

No

67 Replay Timer Timeout Status Set Incorrectly

Description

If a device sends repeated NAKs on the PCIe link to the chipset resulting in a REPLAY_NUM rollover event, the Replay Timer Timeout Status register bit is set incorrectly. The REPLAY_NUM Rollover Status register is set as expected.

Potential Effect on System

Error diagnostic software may misdiagnose the cause of a REPLAY_NUM rollover event on the PCIe link. Software may interpret that the REPLAY_NUM rollover event was caused by the absence of ACKs or NAKs when NAKs were received.

Suggested Workaround

None

Fix Planned

No

68 Error Containment – HyperTransport Interface

Description

A DMA write transaction from the PCIe interface affected by an internal parity error may be presented on the Hypertransport link before the link is transitioned to the syncflood state and the system is halted.

Potential Effect on System

None expected under normal conditions. The corrupted data is not expected to escape to an I/O device before the system is halted. For the specific case of peer-to-peer writes from PCIe crossing from one chipset to another through the processor complex in a multi-chipset environment, corrupted data may reach the destination peer device under specific loading conditions even if internal parity errors in the chipset are configured to halt the system.

Suggested Workaround

None required for most usage scenarios.

Multi-chipset systems utilizing peer-to-peer writes routed between chipsets that require maximum error containment on those transfers (for example, due to the use of non-volatile storage or external I/O interfaces on the destination peer device) may implement a board-level workaround where any chipset's SERR_FATAL# pin output is used to drive the SYNCFLOODIN# input pins on all chipsets within a delay of 20ns. Internal parity errors should be classified as fatal inside each chipset and configured to trigger the SERR_FATAL# pin. Additionally, the SYNCFLOODIN# pin on all chipsets must be enabled.

Fix Planned

No

71 Loss of Flow Control During LDTSTOP# When Using Hypertransport Gen1 Mode

Description

Under highly detailed and specific internal timing conditions, the chipset may fail to properly release flow-control credits to the processor during a HyperTransport Gen1 disconnect sequence due to an LDTSTOP# assertion.

Potential Effect on System

Progressive loss of flow control buffers on the HyperTransport link from the processor to the chipset will eventually result in a system hang.

Suggested Workaround

Operate the link between the chipset and the processor in HyperTransport Gen3 mode if LDTSTOP# needs to be asserted, unless the BIOS can assure that the system is quiesced before LDTSTOP# assertion.

Fix Planned

No

72 DMA Request May Be Incorrectly Logged As Unsupported Request In PCIe Error Reporting Registers

Description

Under highly detailed and specific internal timing conditions with PCIe error reporting enabled and PCIE_CORE_ARB:IOC_CONTROL_2[6] (NBMISCIND:0x12 bit 22) set, the chipset may incorrectly log a valid DMA request as an unsupported request in the associated PCIe bridge configuration space. Although logged as an error, the request is otherwise handled normally. A completion with UR status is not generated on the PCIe interface due to this erratum.

Potential Effect on System

A spurious unsupported request may be logged in the PCIe bridge configuration space including both the standard PCIe capability register set and the advanced error reporting register set. The manifestation of this problem is dependent on the error handling implementation of the platform.

Suggested Workaround

System BIOS should clear PCIE_CORE_ARB:IOC_CONTROL_2[6] (NBMISCIND:0x12 bit 22). This change is implemented in CIMx version 1.0.0.5.

Fix Planned

No

73 PCIe Advanced Error Reporting (AER) Header Log May Record Incorrect Information

Description

Upstream MSI interrupt requests that are aborted by the chipset and logged using the PCIe AER feature will be recorded incorrectly in the AER header log. The chipset, which attempts to log the request address after MSI to Hypertransport interrupt format conversion, only correctly records bits 31:12 of the Hypertransport formatted address. Address bits 11:0 in the AER header log are taken from the MSI interrupt request and bits 63:32 of the Hypertransport formatted address are discarded as the header is recorded using a 3DW format.

Potential Effect on System

If upstream MSI interrupts are aborted by the chipset, software may report an inaccurate transaction address for the failed operation.

Suggested Workaround

None

Fix Planned

No

75 Incorrect Check of PCIe ECRC on Specific Device Configurations of x8 and x16 Links

Description

The chipset may incorrectly check PCIe ECRC (End-to-end Cyclic Redundancy Check) for x8 or x16 links connected via a switch to multiple endpoint devices, where at least one endpoint device is not enabled for ECRC generation.

Potential Effect on System

PCIe AER may log a false ECRC error. The manifestation of this problem is dependent on the error handling implementation of the platform.

Suggested Workaround

In the platform BIOS (i.e., outside of CIMx), do not expose the ECRC checking capability on a PCIe core if any of the associated root ports are x8 or x16 and are connected via a switch to one or more endpoint devices that does not support ECRC or if one of the ports behind the switch is hot-pluggable. See section 7.3.5.2 of the AMD SR5690/5670/5650 Register Programming Requirements document (PID 43872) for additional details.

Fix Planned

No

76 Read Completion Timeout with Relaxed Ordering

Description

A starvation condition is created when a Relaxed Ordering enabled read from an endpoint device does not make forward progress in the presence of a persistent stream of upstream writes from a device behind the same PCIe root port.

Potential Effect on System

When this starvation condition lasts longer than the PCIe read completion timeout in the reading device(s), an AER error is reported. This is expected to be a rare event given the common conditions, such as bursts of DMA write traffic from other ports or bursts of A-Link traffic, that will break the starvation. The timeout has not been observed with production software as persistent writes are not common in real-world applications.

Suggested Workaround

Disable Relaxed Ordering by setting NBMiscCfg space IOC_PCIE_DXX_CNTL [2:0] = 0x6 (where XX are the device numbers for PCIe bridge devices 2 through 13) as well as IOC_PCIE_CNTL which is the register for bridge 8 (the Southbridge A-Link bridge). This change is implemented in CIMx version 1.0.1.2.

Fix Planned

No

77 Read Completion Timeout

Description

A starvation condition is created when a DMA read from a PCIe endpoint device does not make forward progress in the presence of a persistent stream of upstream writes from a device behind the same PCIe root port. Within the chipset's DMA pipeline, the DMA read must be immediately followed by a single DMA write from a different PCIe root port for this issue to occur. One or more DMA reads through the same PCIe root port that issued the blocked read request will break the starvation condition. A burst of DMA writes from a different PCIe root port than the blocked read may break the starvation condition depending on the internal timing of the transactions.

Potential Effect on System

During the starvation condition, DMA reads from all devices will be blocked. When this starvation condition lasts longer than the PCIe read completion timeout in the reading device(s), an AER error is reported. The read completion timeout time has been observed to be as low as 8ms in some devices.

Suggested Workaround

Devices that can issue long streams of DMA writes are advised to modify their DMA access patterns to periodically insert DMA reads. The reads do not need to complete before additional writes are issued. Reads need to be inserted at a sufficient rate to ensure that, if the starvation condition is created, the condition will be broken well within the lowest PCIe completion timeout time in the system.

Fix Planned

No

79 IOMMU Event Log Ordering Violation

Description

The chipset does not maintain producer-consumer ordering between the IOMMU event log DMA writes and IOMMU MMIO register read completions. The CPU may read stale or uninitialized event logs from memory when a read response from the event log tail pointer register passes the corresponding event log DMA write. A series or burst of event log DMA writes are normally required for this issue to occur.

Potential Effect on System

Software may process an event log before it has been completely written, possibly resulting in the operating system or Hypervisor taking improper corrective actions.

Suggested Workaround

The IOMMU driver of the Hypervisor or operating system should initialize the event log buffer to all zeros and write event log entries to zero after they are processed. If software subsequently observes an all zero event log entry, it should re-read the buffer until a non-zero event log is returned.

Fix Planned

No

80 HTIU False Parity Errors

Description

After a cold reset, the chipset may report false parity errors in the HTIU block. Uninitialized internal HTIU buffers may be read if there are outstanding packets on the HyperTransport link after link retraining due to an LDTSTOP or retry event. When the uninitialized data is parity checked, false parity errors may be reported in bits 7:4 of HTIUNBCFG:PARITY_ERROR_STATUS_0 (HTIUNBIND:0x81). Although parity errors may occur, the uninitialized data is not sent out on the link.

The primary chipset is not susceptible to this issue and exposure is limited to the chipsets that are not attached to the Southbridge until their HTIU buffers are initialized under normal functional use via multiple 64B host reads or DMA writes.

Potential Effect on System

False parity errors may be reported.

Suggested Workaround

BIOS should initialize the HTIU buffers prior to enabling parity errors. This change is implemented in CIMx version 1.0.1.7.

Fix Planned

No